Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence

Laurent Kneip[1], Agostino Martinelli[2], Stephan Weiss[1], Davide Scaramuzza[1] and Roland Siegwart[1]
[1] *Autonomous Systems Lab ETH Zurich, Switzerland*
[2] *INRIA, Grenoble - Rhône-alpes, France*

Abstract— **This paper presents a closed-form solution for metric velocity estimation of a single camera using inertial measurements. It combines accelerometer and attitude measurements with feature observations in order to compute both the distance to the feature and the speed of the camera inside the camera frame. Notably, we show that this is possible by just using three consecutive camera positions and a single feature correspondence. Our approach represents a compact linear and multirate solution for estimating complementary information to regular essential matrix computation, namely the scale of the problem. The algorithm is thoroughly validated on simulated and real data and conditions for good quality of the results are identified.**

I. Introduction

Today's palette of robotic applications using visual sensors for Simultaneous Localization and Mapping (SLAM) is steadily growing, and this for obvious reasons. Alternatives such as ultrasonic sensors, planar laser rangefinders, or time-of-flight cameras are sparse in information content, bulky, or inaccurate. The ratio between the information content given by ordinary cameras and the corresponding sensor size or weight is unmatched by any other sensor type. Especially, compact solutions such as small inspection robots or Micro Aerial Vehicles (MAV) tend towards using vision more and more. Recently, Blösch et al. [1] succeeded in the implementation of autonomous MAV navigation in unstructured environments using monocular vision as the only exteroceptive sensor.

The obvious problem with using vision is that cameras only provide a bearing information about features, no depth information. The latter may be recovered by triangulating matched features from multiple views [2], more generally resulting in a visual odometry [3] or visual SLAM approach [4, 5]. However, although these monocular vision algorithms try to conserve a certain scale via relative scale propagation, the output such as camera speed and 3D structure is still only computed up to an unknown scale. This does not represent a serious problem for Augmented Reality (AR) applications, but it certainly does when the control stability of a real robot depends on it, as it is the case for the example mentioned earlier. This paper presents a deterministic solution for finding the absolute-scale velocity of the camera.

An alternative way of measuring the absolute scale is given with stereovision approaches where the metrics is defined via a known baseline. Achtelik et al. [6], for instance, use a stereovision rig for indoor MAV navigation. However, classical scale-variant stereovision concepts fail when the requirements regarding the size of the baseline increase, namely when operating in a large-scale environment with more distant features. Nützi et al. [7] propose to merge the output of a scale-invariant monocular SLAM algorithm with IMU (Inertial Measurement Unit) measurements in an EKF filter holding the scale as an additional variable in the state. However, their approach does not represent a direct computation of scale and speed but rather depends on the reliability of an external SLAM implementation and a critical convergence of the filter. Armesto et al. [8, 9] also address the fusion of inertial measurements and visual pose computations, but rather focus on multirate aspects and assume the position measurements to be correctly scaled.

An integral solution for obtaining also the speed is given by Gemeiner et al. [10], where structure-from-motion and motion estimation filtering based on inertial readings are combined to obtain the absolute scale. The scale for the vision part, however, has to be initialized with a given set of known 3D features. Integral EKF-SLAM-based solutions are given by Eade and Drummond [11] and Strelow and Singh [12] via a replacement of the motion model by motion estimation based on inertial measurements. Kelly and Sukatme [13] demonstrate that a similar approach may even be used for simultaneous camera-to-IMU calibration. The computational complexity, however, is considerable and all the mentioned solutions involve complete SLAM and, thus, do not scale very well with the map-size. Mourikis and Roumeliotis [14] improve the scalability by disregarding older features, hence keeping the complexity of the algorithm constant and linear in function of the number of considered features. Huster et al. [15, 16] finally present a statistical solution to the velocity estimation problem that has similar complexity to the present one. It also uses only one distinctive feature and tries to estimate its relative position in Cartesian space. The drawback with most Kalman filter approaches is that the number of iterations until the filter converges is undefined. The reinitialization of state-variables—if having to switch the observed feature—represents another issue. Another approach is given by Roumeliotis et al. [17] who only consider pairwise images for visual odometry, and fuse the output afterwards with inertial measurements in an EKF. However, their work uses an additional altimeter for solving the unknown scale of the vision algorithm. Baldwin et al. [18] develop a non-linear observer for 6-DOF pose estimation in function of only monocular vision and inertial readings, however, they worked only in simulation and with a static set of features in the field of view.

The main contribution of the present paper consists in a closed-form solution of the velocity determination problem engaging inertial readings and a single feature observation over the past three camera observation points. To the best of our knowledge, this is the first time that such a solution is provided and proven to be working on real data. Furthermore, the fact that only a single feature needs to be considered allows us to design very efficient and robust computation methods such as 1-point RANSAC [19], hence reducing the critical part of the implementation to short-term integration of inertial readings.

The structure of the paper is the following: Section II shows the derivation of the linear solution and the final multirate algorithm; Sections III and IV present evaluation on simulated and real data, respectively; finally Section V concludes the work.

II. Theory

A. Definitions

As a convention, variables are indicated in italic and vectors in boldface throughout the whole paper. Vectors in form of imaginary quaternions are added a tilde, e.g. $\tilde{\mathbf{v}}$. Most of the theoretical derivations are accomplished in the camera frame $Oxyz$ shown in Figure 1. x and y are parallel to the image plane, and z points to the front of the camera. The inertial frame is denoted with $IXYZ$. The camera velocity $\mathbf{v}$ and the acceleration $\mathbf{a}$ are by default expressed in the camera frame. The orientation of the camera frame is expressed with respect to the inertial frame, and given with the Tait-Bryan angles roll φ, pitch θ, and yaw ψ. Alternatively, the attitude may be expressed by a quaternion $\mathbf{q}$. The sampling rate of the IMU is denoted with T.

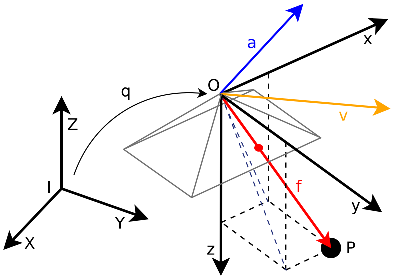


Fig. 1. Acceleration $\mathbf{a}$ and velocity $\mathbf{v}$ of the camera frame $Oxyz$, and feature vector $\mathbf{f}$.

On the vision side, a feature is represented by the feature vector $\mathbf{f}$. Assuming that the camera is calibrated, we can use the normalized image coordinates x_{img} and y_{img}, and $\mathbf{f}$ is then defined by

$$\mathbf{f} = z \cdot \mathbf{f}' = z \cdot \begin{pmatrix} x_{\text{img}} \\ y_{\text{img}} \\ 1 \end{pmatrix}. \tag{1}$$

B. Assumptions

We assume that the camera and the IMU are calibrated, that is, both their intrinsic and extrinsic parameters are known (this is possible by using off-the-shelf toolboxes like [20, 21]). Furthermore, for the moment we assume to have an ideal IMU, that is, gravity and bias free acceleration and gyroscopic measurements with respect to the IMU frame. Knowing the IMU calibration parameters, the inertial readings may be rotated into the camera frame, hence allowing us to elaborate the following theory inside the camera frame only. Finally, we assume to have a reliable feature detector able to match all features visible in each of the last three frames. Noise effects will be shown in Section III.

C. Velocity determination

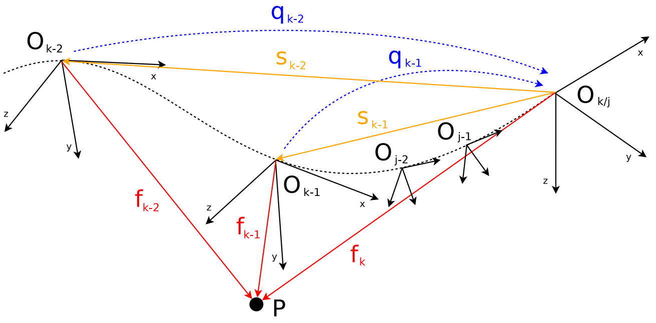


Fig. 2. Three consecutive feature observation points.

Let us consider a moving camera as shown in Figure 2. Each reference frame represents a position where an image or inertial readings have been captured (indices k and j, respectively). The basic idea of the velocity determination consists in fitting its parameters and, at the same time, the feature distance such that—respecting the equations of motion of the camera—the normalized feature coordinates inside the image plane in the previous observation points correspond to their actually measured values. If the last three camera observation points O_k, O_{k-1}, and O_{k-2} happened at t_k, t_{k-1}, and t_{k-2}, and if the relative displacements of the two previous camera frames with respect to O_k are denoted by $\tilde{\mathbf{s}}_{k-1}$ and $\tilde{\mathbf{s}}_{k-2}$, the feature vectors inside the previous frames are given by

$$\begin{aligned} \tilde{\mathbf{f}}_{k-1} &= \mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k - \tilde{\mathbf{s}}_{k-1})\mathbf{q}_{k-1} \\ \tilde{\mathbf{f}}_{k-2} &= \mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k - \tilde{\mathbf{s}}_{k-2})\mathbf{q}_{k-2}, \end{aligned} \tag{2}$$

where quaternions $\mathbf{q}_{k-2}$ and $\mathbf{q}_{k-1}$ represent the relative change in orientation from the observation point O_{k-2} to O_k and from O_{k-1} to O_k, respectively.

The key step now consists in expressing the relative displacements $\tilde{\mathbf{s}}_{k-1}$ and $\tilde{\mathbf{s}}_{k-2}$ in function of the current velocity inside the camera frame and the integration of the acceleration between t_{k-2} and t_k. Since the IMU is typically sampled at a higher rate, we switch to a different time index, namely t_j. If we denote relative displacements with respect to the frame $O_{j-1}xyz$ with $\tilde{\mathbf{s}}_{\dots,j-1}$ and adopt the same notation for velocities, we have (acceleration is always expressed in the current camera frame)

$$\begin{aligned}
\tilde{\mathbf{s}}_{j,j-1} &= \tilde{\mathbf{v}}_{j-1,j-1}T + \frac{1}{2}\tilde{\mathbf{a}}_{j-1}T^2 && (3)\\
\tilde{\mathbf{v}}_{j,j-1} &= \tilde{\mathbf{v}}_{j-1,j-1} + \tilde{\mathbf{a}}_{j-1}T && (4)\\
\tilde{\mathbf{s}}_{j-1} &= \mathbf{q}_{j-1}(-\tilde{\mathbf{v}}_{j-1,j-1}T - \frac{1}{2}\tilde{\mathbf{a}}_{j-1}T^2)\mathbf{q}^*_{j-1} && (5)\\
\tilde{\mathbf{v}}_{j} &= \mathbf{q}_{j-1}(\tilde{\mathbf{v}}_{j-1,j-1} + \tilde{\mathbf{a}}_{j-1}T)\mathbf{q}^*_{j-1} && (6)\\
\Rightarrow \mathbf{q}^*_{j-1}\tilde{\mathbf{v}}_{j}\mathbf{q}_{j-1} &= \tilde{\mathbf{v}}_{j-1,j-1} + \tilde{\mathbf{a}}_{j-1}T && (7)\\
\Rightarrow \quad \tilde{\mathbf{v}}_{j-1,j-1} &= \mathbf{q}^*_{j-1}\tilde{\mathbf{v}}_{j}\mathbf{q}_{j-1} - \tilde{\mathbf{a}}_{j-1}T, && (8)
\end{aligned}$$

and by replacing (8) in (5), we obtain

$$\tilde{\mathbf{s}}_{j-1} = -\tilde{\mathbf{v}}_{j}T + \frac{1}{2}\mathbf{q}_{j-1}\tilde{\mathbf{a}}_{j-1}\mathbf{q}^*_{j-1}T^2. \tag{9}$$

Hence, we also have

$$\begin{aligned}
\tilde{\mathbf{s}}_{j-2,j-1} &= -\tilde{\mathbf{v}}_{j-1,j-1}T + \frac{1}{2}\mathbf{q}_{j-2}\tilde{\mathbf{a}}_{j-2}\mathbf{q}^*_{j-2}T^2 && (10)\\
\tilde{\mathbf{s}}_{j-2} &= \tilde{\mathbf{s}}_{j-1} + \mathbf{q}_{j-1}\tilde{\mathbf{s}}_{j-2,j-1}\mathbf{q}^*_{j-1}\\
&= -\tilde{\mathbf{v}}_{j}T + \frac{1}{2}\mathbf{q}_{j-1}\tilde{\mathbf{a}}_{j-1}\mathbf{q}^*_{j-1}T^2 +\\
\mathbf{q}_{j-1} \quad (- &\quad \tilde{\mathbf{v}}_{j-1,j-1}T + \frac{1}{2}\mathbf{q}_{j-2}\tilde{\mathbf{a}}_{j-2}\mathbf{q}^*_{j-2}T^2)\mathbf{q}^*_{j-1}, && (11)
\end{aligned}$$

and replacing (8) in (11), we obtain

$$\begin{aligned}
\tilde{\mathbf{s}}_{j-2} &= -\tilde{\mathbf{v}}_{j}2T + \frac{3}{2}\mathbf{q}_{j-1}\tilde{\mathbf{a}}_{j-1}\mathbf{q}^*_{j-1}T^2 +\\
&\quad \frac{1}{2}\mathbf{q}_{j-1}\mathbf{q}_{j-2}\tilde{\mathbf{a}}_{j-2}\mathbf{q}^*_{j-2}\mathbf{q}^*_{j-1}T^2. && (12)
\end{aligned}$$

It can be easily proved that (9) and (12) follow a recursive rule, namely that

$$\tilde{\mathbf{s}}_{j-i} = -\tilde{\mathbf{v}}_{j}iT + \tilde{\alpha}_{j\leftarrow j-i},$$

with

$$\tilde{\alpha}_{j\leftarrow b} = \begin{cases} \mathbf{q}_{j-1}(\frac{2(j-b)-1}{2}\tilde{\mathbf{a}}_{j-1}T^2 + \tilde{\alpha}_{j-1\leftarrow b})\mathbf{q}^*_{j-1} & \text{if } j > b\\ 0 & \text{if } j \leq b \end{cases}. \tag{13}$$

The recursive formulation provides one fundamental advantage. Notably, it allows us to sample the IMU readings at a different rate than the camera. This way, the results from an image processing algorithm do no longer have to arrive at constant intervals, that is, we can simply continue until the feature detector provides new interest points.

Splitting up the velocity using the imaginary quaternions $\tilde{\mathbf{q}}_x = (0\ 1\ 0\ 0)^t$, $\tilde{\mathbf{q}}_y = (0\ 0\ 1\ 0)^t$ and $\tilde{\mathbf{q}}_z = (0\ 0\ 0\ 1)^t$, we obtain

$$\tilde{\mathbf{s}}_{j-i} = -(\tilde{\mathbf{q}}_x v_{x,j} + \tilde{\mathbf{q}}_y v_{y,j} + \tilde{\mathbf{q}}_z v_{z,j})iT + \tilde{\alpha}_{j\leftarrow j-i}. \tag{14}$$

Assuming that $t_k = t_j$ and that the past two observation points happened at $t_{k-1} = t_{j-l}$ and $t_{k-2} = t_{j-m}$, we may then reformulate (2) in order to become

$$\begin{aligned}
\tilde{\mathbf{f}}_{k-1} &= \mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT\\
&\quad -\tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\\
\tilde{\mathbf{f}}_{k-2} &= \mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT\\
&\quad -\tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}. && (15)
\end{aligned}$$

This is equivalent to stating that there are m IMU integration points between the second last and the present camera observation point, and l integration points between the last and the present camera observation point. Through (1), we have

$$x_{img,k-1} = \frac{z_{k-1}\cdot x_{img,k-1}}{z_{k-1}} = \frac{\left[\tilde{\mathbf{f}}_{k-1}\right]_x}{\left[\tilde{\mathbf{f}}_{k-1}\right]_z} \tag{16}$$

$$= \frac{\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_x}{\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_z}$$

$$y_{img,k-1} = \frac{z_{k-1}\cdot y_{img,k-1}}{z_{k-1}} = \frac{\left[\tilde{\mathbf{f}}_{k-1}\right]_y}{\left[\tilde{\mathbf{f}}_{k-1}\right]_z} \tag{17}$$

$$= \frac{\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_y}{\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_z}$$

$$x_{img,k-2} = \frac{z_{k-2}\cdot x_{img,k-2}}{z_{k-2}} = \frac{\left[\tilde{\mathbf{f}}_{k-2}\right]_y}{\left[\tilde{\mathbf{f}}_{k-2}\right]_z} \tag{18}$$

$$= \frac{\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_x}{\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_z}$$

$$y_{img,k-2} = \frac{z_{k-2}\cdot y_{img,k-2}}{z_{k-2}} = \frac{\left[\tilde{\mathbf{f}}_{k-2}\right]_y}{\left[\tilde{\mathbf{f}}_{k-2}\right]_z} \tag{19}$$

$$= \frac{\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_y}{\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}}'_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_z},$$

thus four equations with four unknowns, namely $v_{x,k}$, $v_{y,k}$, $v_{z,k}$ and z_k. The used operation $[\tilde{\mathbf{x}}]_x$ is defined as extracting the coordinate x from an imaginary quaternion $\tilde{\mathbf{x}}$. Following the transformations presented in Appendix A, we finally obtain

$$\mathbf{A}\cdot\begin{pmatrix} v_{x,k}\\ v_{y,k}\\ v_{z,k}\\ z_k \end{pmatrix} = \mathbf{b} \Rightarrow \begin{pmatrix} v_{x,k}\\ v_{y,k}\\ v_{z,k}\\ z_k \end{pmatrix} = \mathbf{A}^{-1}\cdot\mathbf{b}, \tag{20}$$

with $\mathbf{A}$ and $\mathbf{b}$ as indicated in Figure 3. Computing the 4×4 matrix $\mathbf{A}$ and the 4×1 vector $\mathbf{b}$ and taking the inverse of $\mathbf{A}$ essentially solves the velocity determination problem using

$$\mathbf{A} = \begin{pmatrix} \left(x_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-1}\right]_{x}\right) lT & ..\tilde{\mathbf{q}}_{y}.. & ..\tilde{\mathbf{q}}_{z}.. & x_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-1}\right]_{x} \\ \left(y_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-1}\right]_{y}\right) lT & ..\tilde{\mathbf{q}}_{y}.. & ..\tilde{\mathbf{q}}_{z}.. & y_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-1}\right]_{y} \\ \left(x_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-2}\right]_{x}\right) mT & ..\tilde{\mathbf{q}}_{y}.. & ..\tilde{\mathbf{q}}_{z}.. & x_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-2}\right]_{x} \\ \left(y_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{q}}_{x}\mathbf{q}_{k-2}\right]_{y}\right) mT & ..\tilde{\mathbf{q}}_{y}.. & ..\tilde{\mathbf{q}}_{z}.. & y_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\mathbf{f}}_{k}'\mathbf{q}_{k-2}\right]_{y} \end{pmatrix}$$

$$\mathbf{b} = \begin{pmatrix} x_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_{x} \\ y_{img,k-1}\left[\mathbf{q}_{k-1}^{*}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_{z} - \left[\mathbf{q}_{k-1}^{*}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_{y} \\ x_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_{x} \\ y_{img,k-2}\left[\mathbf{q}_{k-2}^{*}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_{z} - \left[\mathbf{q}_{k-2}^{*}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_{y} \end{pmatrix}.$$

Fig. 3. Matrix $\mathbf{A}$ and vector $\mathbf{b}$ as required by the velocity determination algorithm. $..\tilde{\mathbf{q}}_{y}..$ equals to its term to the left except that $\tilde{\mathbf{q}}_{x}$ is replaced with $\tilde{\mathbf{q}}_{y}$ (similar is valid for $..\tilde{\mathbf{q}}_{z}..$).

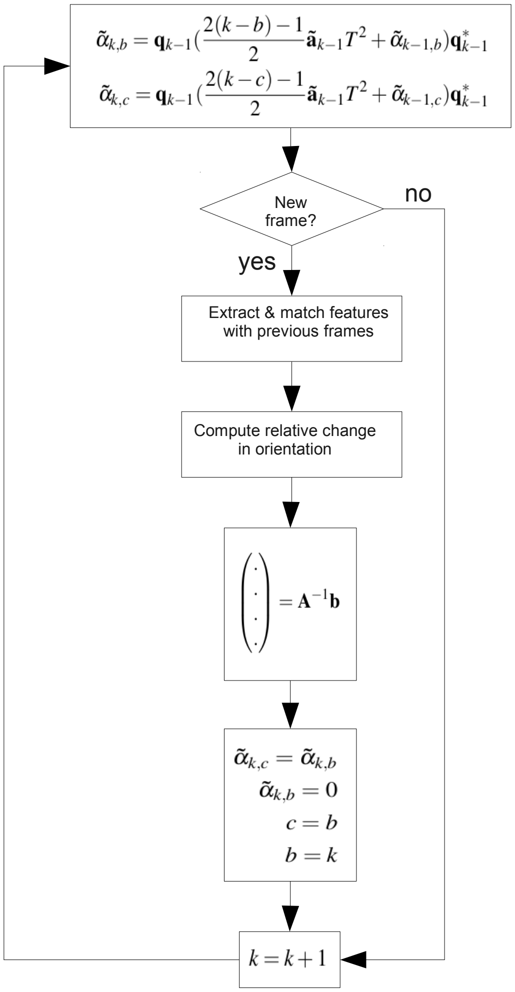
only one single feature. We can observe two advantages: First, $\mathbf{A}$ and $\mathbf{b}$ have a high level of redundancy. This is good because it simplifies the composition of the matrix $\mathbf{A}$, thus leading to reduced computational load. Second, the method depicted here involves only the knowledge about the relative change in orientation, which may be obtained through integration of angular rate sensors, an information that is commonly delivered by IMUs and not too much affected by drift (at least not during smaller integration times). As an alternative, the relative rotational displacement might also be computed via the essential matrix. Hence, the complete problem may be solved via exclusive use of sensorial information directly given in the camera frame. An absolute orientation estimation is however needed in order to remove the gravity portion from the accelerometer values. Note that even though at least three observation points are necessary, it is possible to compute a new velocity for each new camera image, as described in Figure 4.

In order to solve the problem using multiple features at once, it is possible to combine the observation of multiple features into a single sparse matrix $\mathbf{A}$. This, however, leads to increased computational complexity, and 1-point RANSAC [19] to prune false matches and outliers in the speed computation is definitely preferable.

III. Simulation results

The theories established in Section II will now be evaluated in simulation. The camera is assumed to be down-looking with an opening angle of 180°. The idea behind the orientation is that the camera is mounted on a quadrotor. It is then equivalent to the helicopter attitude and hence related to the translational accelerations via a simple point-mass model. The features are placed on a flat surface at an average distance of 5 m below the camera. They are evenly distributed and cover an area of 10×10 m^2. The sampling frequencies are assumed to be 100 Hz for the IMU and 10 Hz for the camera. The duration of each experiment is 30 s.

The first experiment addresses the observation of a single feature located around the center of the considered area.



$\tilde{\alpha}_{k,b} = \mathbf{q}_{k-1}(\frac{2(k-b)-1}{2}\tilde{\mathbf{a}}_{k-1}T^2 + \tilde{\alpha}_{k-1,b})\mathbf{q}_{k-1}^{*}$
$\tilde{\alpha}_{k,c} = \mathbf{q}_{k-1}(\frac{2(k-c)-1}{2}\tilde{\mathbf{a}}_{k-1}T^2 + \tilde{\alpha}_{k-1,c})\mathbf{q}_{k-1}^{*}$
New frame?
no
yes
Extract & match features with previous frames
Compute relative change in orientation
$\begin{pmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} = \mathbf{A}^{-1}\mathbf{b}$
$\tilde{\alpha}_{k,c} = \tilde{\alpha}_{k,b}$
$\tilde{\alpha}_{k,b} = 0$
$c = b$
$b = k$
$k = k+1$

Fig. 4. IMU callback function executed at a period of T and providing a velocity update each time a new camera frame has been captured. Note that the buffering of orientation quaternions and feature angles/descriptors is omitted. Also note that the variables l and m in Eq. 20 need to be replaced by $k-b$ and $k-c$ using the terminology here.

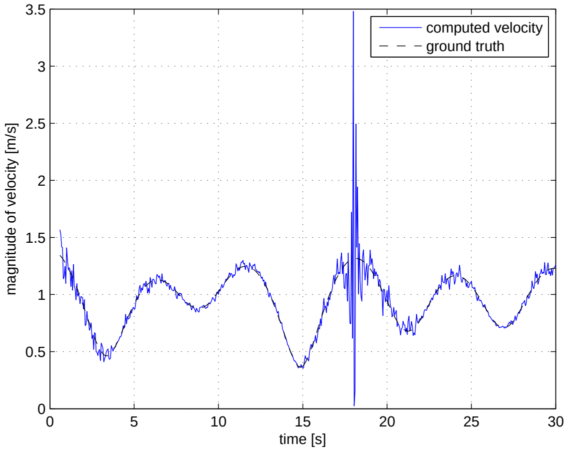


Fig. 5. Absolute error of velocity determination during the tracking of a single feature.

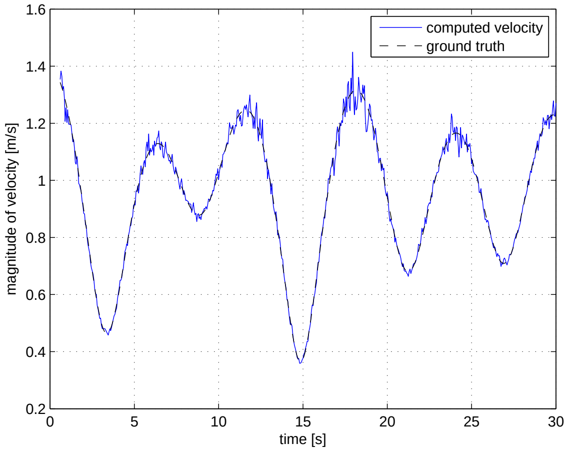


Fig. 6. Absolute error of velocity determination during the observation of many features.

White Gaussian noise with a standard deviation of $0.1\frac{m}{s\sqrt{h}}$[1] is added to the acceleration values, a value taken from a real IMU. The result of the velocity determination using a single feature is shown in Figure 5. The RMS value of the estimation error is $0.142\ \frac{m}{s}$ for an average ground-truth absolute velocity of $0.948\ \frac{m}{s}$, hence resulting in a relative error of about 15%. Furthermore, it can be observed that the precision of the measurement gets reduced around $t = 18s$. This happens because the considered feature has a very small displacement in the image plane over the last three frames. The motion is then not observable through this feature and $\mathbf{A}$ becomes close to singular. For instance, this is the case if the considered feature is perfectly aligned with the direction of motion. When this happens, a solution is given by the consideration of a different feature at a different position in the image plane.

As shown in our next experiment, an alternative is given by the application of 1-point RANSAC to all the features observable in each of the last three frames. The results are shown in Figure 6. It can be observed that the results in the critical region around $t = 18s$ are significantly improved. Furthermore, we obtain a general reduction of the RMS value of the error down to only $0.023\ \frac{m}{s}$, hence a relative error of 2.5%. Moreover, even if all the features are considered, the computation stays efficient due to the fact that each RANSAC hypothesis is instantiated using only one single feature.

Another obvious limitation is when the result of the integration of the acceleration values over the last three camera observation points is very small, which happens, for instance, when the velocity of the camera is constant. The acceleration values then remain zero and hence the motion unobservable. This event may be observed in our next experiment—shown in Figure 7—at $t = 21s$. It has been verified that at this instant the integration of the acceleration over the three frames is almost zero, which leads to a singular matrix for every observed feature and thus a non-observable scale for the relative displacement of the camera. Note that both critical cases can be easily detected at runtime by analyzing the condition of $\mathbf{A}$ and the result of the IMU integration.

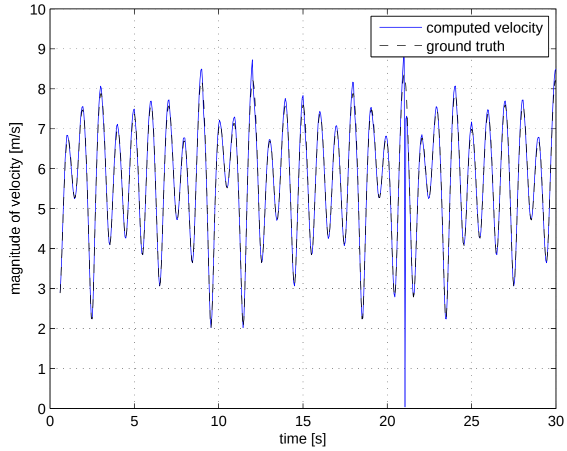


Fig. 7. Absolute error of velocity determination during the observation of multiple features, this time with increased motion dynamics.

Another important question is to what extend the average size of the magnitude of the acceleration influences the precision of the speed determination. Therefore, the motion dynamics and velocities in our previous experiment (Figure 7) have been increased to an average value of $5.738\ \frac{m}{s}$, and the resulting RMS value for the error is $0.3558\ \frac{m}{s}$. Thus, the relative error increased to 6%. This can be explained by the fact that large accelerations also cause higher velocities, and hence also a more dynamic optical flow in the image plane. The camera observation points being triggered by

[1] $\frac{m}{s\sqrt{h}}$ is the standard unit to express the error of an IMU (m is meters, s is seconds, and h is hours)

thresholded disparity, the IMU integration times and thus also noise cancelling effects are automatically getting smaller. A similar increase of the relative error can be observed for too small motion dynamics, where the noise of the accelerometer signal then outweights the actual signal and high noise integration times cause significant errors. It can be concluded that the reliability of the algorithm depends on good characteristics of the motion, which means not too short and not too long integration times of the IMU data.

IV. Experiments on Real Data

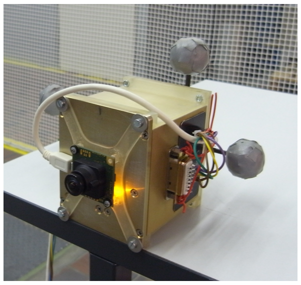


Fig. 8. Crossbow-IMU with mounted uEye camera and tracking balls for the motion capturing system.

In order to have a practical evaluation of the algorithm introduced in Section II, a real dataset has been collected with the setup presented in Figure 8. It consists of a uEye UI-122xLE—a small monochrome USB-camera gathering 752×480 images with global shutter at a rate of 15 Hz—and a Crossbow VG400CC-200 IMU providing measurement updates at a rate of 75 Hz. According to the datasheet, the noise contained in the acceleration measurements amounts to $0.5\frac{m}{s\sqrt{h}}$. Figure 9 shows an image of the environment captured by the camera itself during the experiment. The field of view of the camera is 150°, and the latter is calibrated using the omnidirectional camera calibration toolkit by Scaramuzza [20]. The extrinsic calibration of the IMU is realized using the camera-inertial calibration toolkit by Lobo [21]. Reliable ground-truth data has been obtained by conducting the experiments at the ETH Zurich Flying Machine Arena [22], which is equipped with a Vicon motion capture system. Therefore, the sensor-setup is equipped with three additional tracking balls. Synchronization between visual, inertial, and ground-truth data has been obtained in a pre-processing step. A Visual-SLAM implementation [4] has been applied to the captured image sequence in order to derive rotational velocities. The time-shifts have then been obtained by maximizing the correlation between the rotational velocities from all three sensors (camera, IMU, and Vicon system).

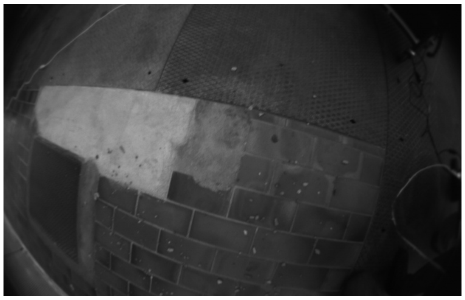


Fig. 9. Image of the environment captured by the uEye camera.

The challenge when using a real IMU obviously consists in subtracting the gravity vector from the acceleration measurements, which can only be done if knowledge about the absolute orientation is available. Therefore, the latter has been recovered by fusing the accelerometer and rate-gyro measurements in a standard complementary attitude filter, similar to the one presented by Baerveldt [23]. After back and forth rotation and removal of the gravity influence, the preconditioning of the acceleration measurements is concluded with the removal of the bias. The bias is safely estimated by simply taking the average of the acceleration values in all three directions over an extended period of time. This works well since the biases are mostly constant and only varying as a function of the temperature. Please note that the prefiltering does not depend on the translational velocity and can hence be started beforehand in order to deliver stable orientation information for the velocity determination algorithm upon convergence. The determined velocity can then still be used for initializing a translational motion estimation filter.

On the computer vision side, the implementation is engaging the SIFT feature detector by Lowe [24]. A future real-time implementation would obviously suggest the use of a faster detector, like, for instance, the FAST feature detector by Rosten [25].

The result of an over 35s long period with sufficient motion dynamics is shown in Figure 10. The mean value of the velocity is 0.3889 $\frac{m}{s}$ and the average of the velocity estimation error is 0.1447 $\frac{m}{s}$. The relative error thus results to 37%. The increase of the relative error compared to the

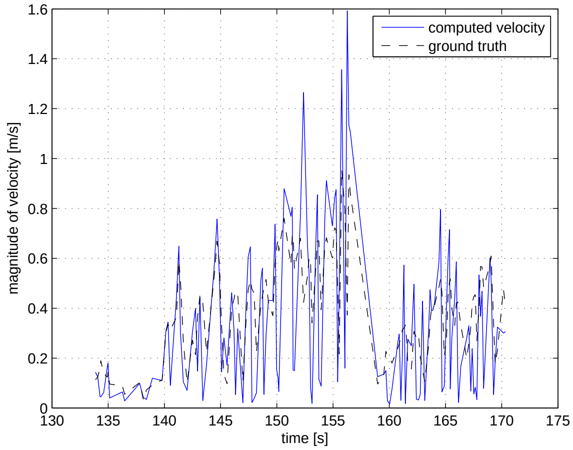


Fig. 10. Absolute error of velocity determination on real data.

results obtained in simulation is related on one hand to the higher noise component in the inertial readings, and on the other hand to temporarily missing dynamics of the camera motion. Nevertheless, a visual odometry implementation with relative scale propagation could still benefit from our method, especially for setting the absolute scale whenever the conditions imposed by the motion dynamics are favorable. Considering the bounded scale drift, the scale factor can as well be simply averaged over a longer period. We can conclude that our algorithm represents an important and very efficient contribution to estimate the absolute scale and velocity in monocular visual odometry.

V. Conclusions and outlook

In this paper, we povided a closed-form solution for velocity determination combining visual and inertial information, which requires only a single feature observation across three views. This approach operates at camera framerate in discrete time-space and uses only information retrieved from the moving camera/IMU frame. It is a multirate approach, which means that the higher sampling rate of the IMU is also supported. The fact that our method requires as few as a single feature correspondence means that multiple correspondences can be very efficiently merged with a 1-point RANSAC approach. The presented algorithm returns complementary information to pure epipolar geometry computation and solves for metric camera velocity and metric feature distance, hence the scale of the problem.

While standard visual odometry provides a reliable speed estimation up to a slowly drifting scale factor, the method presented in this paper provides an estimation at an absolute scale. Future research investigates the optimal combination of this complementary information for a robust real-time camera velocity estimation in absolute scale.

VI. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly) and from the Swiss National Science Foundation under grant agreement n. 200021 125017/1.

References

[1] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.

[2] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003.

[3] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 1, pages 652–659, 2004.

[4] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, 2007.

[5] A. Davison, D. Reid, D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):1052–1067, 2007.

[6] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In *Proceedings of The SPIE*, volume 7332, Orlando, FL, USA, 2009.

[7] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of Intelligent and Robotic Systems*, 61(1–4):287–299, 2011.

[8] L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *Int. J. Rob. Res.*, 26(6):577–589, 2007.

[9] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. In *Proceedings of The IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004.

[10] P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous motion and structure estimation by fusion of inertial and vision data. *The International Journal of Robotics Research*, 26(6):591–605, 2007.

[11] E. Eade and T. Drummond. Scalable monocular SLAM. In *Proceedings of The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–476, Washington, DC, USA, 2006. IEEE Computer Society.

[12] D. Strelow and S. Singh. Online motion estimation from image and inertial measurements. In *Workshop on Integration of Vision and Inertial Sensors (INERVIS)*, Coimbra, Portugal, 2003.

[13] J. Kelly and G. S. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In *Proc. IEEE International Conference on Computational Intelligence in Robotics and Automation*, Korea, 2009.

[14] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Roma, 2007.

[15] A. Huster and S. M. Rock. Relative position estimation for intervention-capable AUVs by fusing vision and inertial measurements. In *Proceedings of The 12th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, 2001.

[16] A. Huster, E. W. Frew, and S. M. Rock. Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements. In *Proceedings of The Oceans Conference*, volume 3, pages 1857–1864, Biloxi, 2002. MTS/IEEE.

[17] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, pages 4326–4333, Washington D.C., 2002.

[18] G. Baldwin, R. Mahony, and J. Trumpf. A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Kobe, 2009.

[19] J. Civera, O.G. Grasa, A. Davison, and J.M.M. Montiel. 1-point RANSAC for EKF-based structure from motion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, 2009.

[20] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[21] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575, 2007.

[22] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Anchorage, 2010.

[23] A.J. Baerveldt and R. Klang. A low-cost and low-weight attitude estimation system for an autonomous helicopter. *Proc. Intelligent Engineering Systems*, pages 391–395, 1997.

[24] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[25] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443, 2006.

APPENDIX A

The following shows the detailed steps in order to derive the linear solution from the four constraints (16), (17), (18), and (19).

$$\left\{\begin{array}{lcl}
\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_z x_{img,k-1} & = & \left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_x \\
\left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_z y_{img,k-1} & = & \left[\mathbf{q}^*_{k-1}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})lT - \tilde{\alpha}_{j\leftarrow j-l})\mathbf{q}_{k-1}\right]_y \\
\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_z x_{img,k-2} & = & \left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_x \\
\left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_z y_{img,k-2} & = & \left[\mathbf{q}^*_{k-2}(z_k\tilde{\mathbf{f}'}_k + (\tilde{\mathbf{q}}_x v_{x,k} + \tilde{\mathbf{q}}_y v_{y,k} + \tilde{\mathbf{q}}_z v_{z,k})mT - \tilde{\alpha}_{j\leftarrow j-m})\mathbf{q}_{k-2}\right]_y
\end{array}\right.$$

$$\Rightarrow\left\{\begin{array}{c}
(\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_z z_k + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_z lTv_{x,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_z lTv_{y,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_z lTv_{z,k} - \left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_z)x_{img,k-1} \\
= \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_x z_k + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_x lTv_{x,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_x lTv_{y,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_x lTv_{z,k} - \left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_x \\[2ex]
(\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_z z_k + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_z lTv_{x,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_z lTv_{y,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_z lTv_{z,k} - \left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_z)y_{img,k-1} \\
= \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_y z_k + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_y lTv_{x,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_y lTv_{y,k} + \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_y lTv_{z,k} - \left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_y \\[2ex]
(\left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_z z_k + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_x\mathbf{q}_{k-2}\right]_z mTv_{x,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_y\mathbf{q}_{k-2}\right]_z mTv_{y,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_z\mathbf{q}_{k-2}\right]_z mTv_{z,k} - \left[\mathbf{q}^*_{k-2}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_z)x_{img,k-2} \\
= \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_x z_k + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_x\mathbf{q}_{k-2}\right]_x mTv_{x,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_y\mathbf{q}_{k-2}\right]_x mTv_{y,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_z\mathbf{q}_{k-2}\right]_x mTv_{z,k} - \left[\mathbf{q}^*_{k-2}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_x \\[2ex]
(\left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_z z_k + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_x\mathbf{q}_{k-2}\right]_z mTv_{x,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_y\mathbf{q}_{k-2}\right]_z mTv_{y,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_z\mathbf{q}_{k-2}\right]_z mTv_{z,k} - \left[\mathbf{q}^*_{k-2}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_z)y_{img,k-2} \\
= \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_y z_k + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_x\mathbf{q}_{k-2}\right]_y mTv_{x,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_y\mathbf{q}_{k-2}\right]_y mTv_{y,k} + \left[\mathbf{q}^*_{k-2}\tilde{\mathbf{q}}_z\mathbf{q}_{k-2}\right]_y mTv_{z,k} - \left[\mathbf{q}^*_{k-2}\tilde{\alpha}_{j\leftarrow j-m}\mathbf{q}_{k-2}\right]_y
\end{array}\right.$$

$$\Rightarrow\left\{\begin{array}{c}
(\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_z x_{img,k-1} - \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_x)\cdot z_k + (\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_z x_{img,k-1} - \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_x\mathbf{q}_{k-1}\right]_x)lT\cdot v_{x,k} + \\
(\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_z x_{img,k-1} - \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_y\mathbf{q}_{k-1}\right]_x)lT\cdot v_{y,k} + (\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_z x_{img,k-1} - \left[\mathbf{q}^*_{k-1}\tilde{\mathbf{q}}_z\mathbf{q}_{k-1}\right]_x)lT\cdot v_{z,k} = \\
\left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_z x_{img,k-1} - \left[\mathbf{q}^*_{k-1}\tilde{\alpha}_{j\leftarrow j-l}\mathbf{q}_{k-1}\right]_x \\[2ex]
(\left[\mathbf{q}^*_{k-1}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-1}\right]_z y_{img,k-1} - \qquad\qquad\qquad \dots \\[2ex]
(\left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_z x_{img,k-2} - \qquad\qquad\qquad \dots \\[2ex]
(\left[\mathbf{q}^*_{k-2}\tilde{\mathbf{f}'}_k\mathbf{q}_{k-2}\right]_z y_{img,k-2} - \qquad\qquad\qquad \dots
\end{array}\right.$$

$$\Rightarrow \mathbf{A}\cdot\begin{pmatrix} v_{x,k} \\ v_{y,k} \\ v_{z,k} \\ z_k \end{pmatrix} = \mathbf{b}$$